New Factorization Techniques and Fast Serial and Parallel Algorithms for Operational Space Control of Robot Manipulators

Amir Fijany†, Karim Djouani‡, George Fried‡, and Jean Pontnau‡

† *Jet Propulsion Laboratory,* California *Institute of Technology*

4800 Oak Grove Drive, Pasadena, CA 91109, USA

Tel: (818) 906-6491, Fax:(818) 306-6912, Email: Amir.Fijany@jpl.nasa.gov

‡ *Laboratoire d'Informatique Industrielle et d'Automatique (LIIA), University Paris XII Creteil, France*

Abstract. In this paper a new factorization technique for computation of inverse of mass matrix, M:', and the operational space mass matrix, A, as arising in implementation of the operational space control scheme, is presented. This technique results in Schur Complement factorization of both M-1 and A and subsequently new O(N) algorithms for their computation. These O(N) algorithms are highly efficient for parallel computation. To our knowledge, they represent the first algorithms that can be fully parallelized resulting in both time- and processor-optimal parallel algorithms. Using these algorithms, the OSC scheme can be implemented with an optimal efficiency in both serial and parallel environment. However, in addition to computational efficiency, these algorithms provides a deeper physical insight into the structure of computation which can be exploited for a better design of task space control schemes.

Key Words: Robot Dynamics and Control, Operational Space Control, Parallel Computation

I. Introduction

Consideration of dynamics is essential in the design, analysis, and control of robot manipulator systems. Most of the proposed approaches to dynamic control are based on the joint-space dynamic models. However, task specification for motion and contact forces, dynamics, and force sensing feedback are closely linked to the End-Effecter (EE), i.e., they are defined in the operational space (the Cartesian task space) of the robot manipulators. Thus, the dynamic behavior of the EE is one of the most significant characteristics in evaluating the performance of robot manipulator systems [1]. The EE dynamic modeling and control is also of particular importance for tasks that involve combined motion and contact forces of the EE.

To allow the description, analysis, and control of manipulator systems with respect to the dynamic characteristics of their EEs, Khatib [1,2] has suggested the Operational Space formulation. This formulation enables the description of both dynamics and control strategies at the EE level. However, the Operational Space Control (OSC) scheme is significantly more computation-intensive that the joint-space dynamic control strategies. The joint-space control schemes require the computation of the inverse dynamics of the manipulator at the joint level, which can be efficiently accomplished by using the O(N) recursive Newton-Euler (N-E) formulation [3]. The OSC scheme, in addition to the inverse dynamics, also requirea the computation of the inverse of Joint-Space Mass Matrix, At-1, which corresponds to the solution of forward dynamics problem, and the Operational Space Mass Matrix, A.

In [4] a recursive O(N) algorithm for computation of A is developed. A recursive O(N) algorithm for computation of Λ^{-1} is presented in [5]. Once A-1 (A) is obtained then A (Λ^{-1}) can be computed by inverting a 6 x 6 matrix with a coat of O(l). The O(N) algorithms in [4,5] along with the O(N) algorithms for forward *dynamics* [6,7] can be used as a set of optimal serial algorithms for implementation of the OSC scheme. However, in order to meet the real-time constraints in the implementation, further significant improvement in the computational efficiency is needed. It is clear that, given the relative maturity of the serial algorithms, any such improvement in the computational efficiency can be only achieved through exploitation of parallelism in the computation.

However, there seems to be no report on the development of efficient parallel algorithms for computation of A and/or Λ^{-1}. The O(N) algorithms in [4,5] result in a set of nonlinear recurrences which are similar to those arising in the O(N) algorithms [6,7] for forward dynamics problem. An extensive analysis of efficiency of these recurrences for parallel computation is presented in [8,9] wherein it has been shown that they are strictly sequential, that is, regardless of the number of processors employed, their computation can be speeded up only by a small constant factor. As a result, the O(N) algorithms in [4,5] are also strictly sequential and cannot be efficiently parallelized.

In this paper, starting with a recently developed factorization of *M-1* in form of Schur Complement, we derive new Schur Complement factorization for Λ^{-1} and Λ. These factorization result in novel algorithms for implementation of OSC scheme with the following advantages over the existing algorithms:

1. Optimal Serial Efficiency: They have an optimal complexity of O(N) for a sequential implementation. More importantly, they are more **efficient** (in terms of number of operations) than the previous algorithm since they exploit a larger degree of synergism between the computation of M^{-1} and Λ^{-1} or Λ.
2. 0 **pimal** Parallel Efficiency: They can be fully **parallelized** leading to time lower bound of $O(Log\ N)$, by using an optimal number of O(N) processors. In addition to such a **theoretical** significance, they are also highly **efficient** for practical Implementation on **MIMD** parallel architectures.
3. Deeper **Physical Insight**: The **factorizations** also provide a deeper **physical insight** into the structure of computation. This property can be further **exploited** to gain a better understanding in the design of control schemes.

In this paper, due to the lack of space, we mainly concentrate ***on*** the mathematical **derivation** of the algorithms and a brief analysis of **their** efficiency for serial and parallel computation. **This** paper is organized as follows. In **§II,** the OSC scheme **is** briefly reviewed and its computational complexity is analyzed. Notation and some preliminaries are presented in **§III.** In **§IV, Schur** Complement factorization of M^{-1} is reviewed. **Schur Complement** factorization of Λ^{-1} and Λ are derived in **§V.** The serial and parallel efficiency of the algorithms are briefly discussed in **§VI.** Finally, some concluding remarks are made in **§VII.**

II. Operational Space Dynamic Formulation

A. Formulation

In this section we **breifly** review the operational space dynamic formulation. More detailed discussion can be found in [1,2]. The manipulator **joint**-space dynamics is given by

$$\mathcal{M}\ddot{Q} + C + G = \Gamma \tag{1}$$

where G(Q) and C(Q, Q) are the **gravitional** and **coriolis/centrifugal forces,** respectively. The operational space dynamics is given by [1,2]

$$\Lambda\dot{V}_{N+1} + \mathcal{C} + \mathcal{G} = F_{N+1} \tag{2}$$

where $\Lambda \epsilon \Re^{6\times 6}$ is the **Operational** Space Maas **Matrix**. The terms $\mathcal{G}$ and $\mathcal{C}$ are the **gravitional** and **coriolis/centrifugal** forces described at the EE level. The spatial force, **velocity,** and acceleration of the EE are related to the joint forces, velocities, and accelerations as follows:

$$\Gamma = \mathcal{J}^t F_{N+1} \tag{3}$$

$$V_{N+1} = \mathcal{J}\dot{Q} \tag{4}$$

$$\dot{V}_{N+1} = \mathcal{J}\ddot{Q} + \dot{\mathcal{J}}\dot{Q} \tag{5}$$

Equation (1) can be written as

$$\ddot{Q} + \mathcal{M}^{-1}(C + G) = \mathcal{M}^{-1}\Gamma \tag{6}$$

Premultiplying Eq. (6) by $\mathcal{J}$ **(assuming** $\mathcal{J}$ is **non**-singular), we get

$$\mathcal{J}\ddot{Q} + \mathcal{J}\mathcal{M}^{-1}(C + G) = \mathcal{J}\mathcal{M}^{-1}\Gamma \tag{7}$$

Substituting **Eqs.** (3) and (5) into Eq. (7) gives

$$\dot{V}_{N+1} + \mathcal{J}\mathcal{M}^{-1}(C + G) - \dot{\mathcal{J}}\dot{Q} = \mathcal{J}\mathcal{M}^{-1}\mathcal{J}^t F_{N+1}$$

$$\begin{aligned}(\mathcal{J}\mathcal{M}^{-1}\mathcal{J}^t)^{-1}\dot{V}_{N+1} + (\mathcal{J}\mathcal{M}^{-1}\mathcal{J}^t)^{-1} \\ (\mathcal{J}\mathcal{M}^{-1}(C + G) - \dot{\mathcal{J}}\dot{Q}) = F_{N+1}\end{aligned} \tag{8}$$

Comparing Eq. (8) with Eq. (2), and distinguishing bet ween **velocit** y-dependent and non **velocity**-dependent terms, **it** follows that

$$\Lambda = (\mathcal{J}\mathcal{M}^{-1}\mathcal{J}^t)^{-1} \Rightarrow \Lambda^{-1} = \mathcal{J}\mathcal{M}^{-1}\mathcal{J}^t \tag{9}$$

$$\mathcal{C} = \Lambda(\mathcal{J}\mathcal{M}^{-1}C - \dot{\mathcal{J}}\dot{Q}) \tag{10}$$

$$\mathcal{G} = \Lambda\mathcal{J}\mathcal{M}^{-1}G \tag{11}$$

Equations (9)-(1 1) describe the relationships **between** the operational space and **joint space quantities**. A decoupled and linearized EE **dynamics** of the form $\dot{V}_{N+1} = u$ can be then obtained by a feedback linearization scheme given by [1]

$$F_{N+1} = \Lambda(u + \mathcal{J}\mathcal{M}^{-1}(C + G) - \dot{\mathcal{J}}\dot{Q}) \tag{12}$$

$$\Gamma = \mathcal{J}^t F_{N+1} \tag{13}$$

B. Computational Complexity Analysis

The algorithms presented in this paper can be used for the evaluation of **Eqs.** (9)-(11). However, such an evaluation is more **suitable for** dynamic **analysis** than control. In the following, we concentrate on the efficient implementation of controller given **by Eqs.** (12)-(13).

The computation of the nonlinear term (C+G) can be achieved by computing the N-E formulation while setting joint accelerations to zero, i.e., with $\dot{Q} = 0$. There have been several reports on the development of the numerical methods for computation of the matrix $\dot{\mathcal{J}}$ (see, for example, [13]). However, only the explicit computation of the vector $\dot{\mathcal{J}}\dot{Q}$ rather than the matrix $\dot{\mathcal{J}}$ is needed. In this sense, based on its physical interpretation, the vector $\dot{\mathcal{J}}\dot{Q}$ can be obtained with a small cost as a by-product of computation of the term (C+G). To see this, note that if in Eq. (5) the vector $\ddot{Q}$ is set the zero then the resulting vector $V'_{N+1} = \dot{\mathcal{J}}\dot{Q}$ represents the EE spatial acceleration due to the joint velocities. Therefore, if the forward recursion in the N-E formulation is slightly modified to compute the spatial acceleration of the EE then, by setting $\ddot{Q} = 0$, both the terms (C+G) and V'_{N+1} can be computed. In fact, as is shown in [12], even if the matrix $\dot{\mathcal{J}}$ is explicitly computed, then its multiplication by the vector $\dot{Q}$ results in a slightly modified forward recursion of the N-E formulation. As a result, by using the N-E formulation the serial computation of the vectors (C+G) and V'_{N+1} can be performed with a cost $O(N)$. As is shown in [14], the computation of the N-E formulation can be fully parallelized and performed in a time of $O(Log\, N)$ by using O(N) processors.

For an efficient implementation of Eq. (12) the operator application of $\mathcal{M}^{-1}$, *i.e.,* its multiplication by a vector which is equivalent to the solution of forward dynamics problem, rather than its explicit computation is needed. By using the algorithm in this paper, such an operator application can be performed in O(N), in a serial fashion, and in $O(Log$ N) with O(N) processors in a fully parallel fashion. Note also, that the explicit computation of $\mathcal{J}$ is not needed since the multiplication of vector by $\mathcal{J}$, in Eq. (12), or $\mathcal{J}^t$, in Eq. (13) can be performed in a recursive fashion involving simple linear recurrences. These recurrences can be computed with a cost of O(N) in a serial fashion and with a cost of $O(Log$ N) by using O(N) processors, in a parallel fashion.

As will be shown, our algorithms allow O(N) serial and $O(Log$ N) with O(N) processors parallel computation of A. This result demonstrates that the OC scheme can be implemented with an optimal serial and, particularly, parallel efficiency.

III. Notation and Preliminaries

A. Spatial and Global Notation

In the following, we use spatial and global notation which allow a compact representation of derivation of various factorization. For the sake of simplicity, only Joints with one revolute DOF are considered here. However, the results can be extended to the joints with different and/or more DOFs.

With any vector V, a matrix $\tilde{V} \epsilon \Re^{3\times 3}$ can be associated whose representation in any frame is a skew symmetric matrix:

$$\tilde{V} = \begin{bmatrix} 0 & -V_z & V_y \\ V_z & 0 & -V_x \\ -V_y & V_x & 0 \end{bmatrix}$$

where V_x, V_y, and V_z are the components of V in the frame considered. The matrix $\tilde{V}$ has the properties that $\tilde{V}^t = -\tilde{V}$ and $\tilde{V}_1 V_2 = V_1 \times V_2$, i.e., it is a vector cross-product operator. A matrix $\hat{V} \epsilon \Re^{6\times 6}$ associated to the vector V is also defined as

$$\hat{V} = \begin{bmatrix} U & \tilde{V} \\ 0 & U \end{bmatrix} \quad \text{and} \quad \hat{V}^t = \begin{bmatrix} U & 0 \\ -\tilde{V} & U \end{bmatrix}$$

where here (and through the rest of the paper) U and O stand for unit and zero matrices of appropriate size. The spatial velocities of two rigidly connected points A and B are related as

$$V_A = \hat{P}^t_{A,B} V_B$$

where $P_{A,B}$ denotes the position vector from B to A. The matrix $\hat{P}_{A,B}$ has the properties as

$$\hat{P}_{A,B}\hat{P}_{B,C} = \hat{P}_{A,C} \quad \text{and} \quad \hat{P}^{-1}_{A,B} = \hat{P}_{B,A} \tag{14}$$

The spatial forces acting at two rigidly connected points A and B are related:

$$F_B = \hat{P}_{A,B} F_A$$

If the linear and angular velocities of point A are zero then

$$\dot{V}_A = \hat{P}^t_{A,B} \dot{V}_B$$

The spatial inertia of link i about point j is denoted by $I_{i,j}$. The spatial inertia of link i about its center of mass is designated as $I_{i,Ci}$. The spatial inertia of link i about point O_i (denoted aa I_i) is obtained aa

$$I_i = \hat{S}_i I_{i,Ci} \hat{S}^t_i \tag{15}$$

Equation (15) represents the *parallel axis theorem* for propagation of spatial inertia.

A bidiagonal block matrix $\mathcal{P} \epsilon \Re^{6N\times 6N}$ is defined as

$$\mathcal{P} = \begin{bmatrix} U & & & & & \\ -\hat{P}_{N-1} & U & & 0 & & \\ 0 & -\hat{P}_{N-2} & U & & & \\ 0 & 0 & & & & \\ \vdots & \vdots & & & & \\ 0 & 0 & & & -\hat{P}_1 & U \end{bmatrix}$$

Note that, according to our notation, $P_{i+1,i} = P_i$.

B. Operator Expression of Jacobian Matrix

Following the treatment in [5], a factorization of Jacobian matrix by using our notation is derived as follows. The velocity propagation for a serial chain of rigid bodies is given by (Fig. 1)

$$V_i - \hat{P}^t_{i-1} V_{i-1} = H_i \dot{Q}_i \qquad (16)$$

which, by using the matrix $\mathcal{P}$, can be expressed in a global form as

$$\mathcal{P}^t \mathcal{V} = \mathcal{H}\dot{Q} \Rightarrow \mathcal{V} = (\mathcal{P}^t)^{-1}\mathcal{H}\dot{Q} \qquad (17)$$

The EE spatial velocity, V_{N+1}, is obtained by writing Eq. (16) for $i = N + 1$ as

$$V_{N+1} - \hat{P}^t_N V_N = 0 \Rightarrow V_{N+1} = \hat{P}^t_N V_N \qquad (18)$$

Defining $\beta = [\hat{P}^t_N, 0, 0, \ldots, 0] \epsilon \Re^{6\times 6N}$, from Eqs. (17)-(18), we get

$$V_{N+1} = \beta \mathcal{V} = \beta (\mathcal{P}^t)^{-1}\mathcal{H}\dot{Q} \qquad (19)$$

Comparing Eqs. (4) and (19), an operator expression (or, a factorization) of Jacobian matrix is then given by

$$\mathcal{J} = \beta(\mathcal{P}^t)^{-1}\mathcal{H} \qquad (20)$$

C. Equations of Motion

The equations of motion given by Eq. (1) can be written as

$$\mathcal{M}\ddot{Q} = \Gamma - C(\dot{Q},Q) - G(Q), \text{ or}$$

$$\mathcal{M}\ddot{Q} = \mathcal{F}_T \;\Rightarrow\; \ddot{Q} = \mathcal{M}^{-1}\mathcal{F}_T \qquad (21)$$

where $\mathcal{F}_T = \text{Col}\{F_{Ti}\} = \Gamma - C(\dot{Q},Q) - G(Q)\ \epsilon\Re^{N\times 1}$ represents the acceleration-dependent component of the control force. From Eq. (21) the multi body system can be assumed as a system at rest which upon application of the control force $\mathcal{F}_T$ accelerates in space. The propagation of accelerations and forces among the links of serial chain are then given by

$$\dot{V}_i = \hat{P}^t_{i-1}\dot{V}_{i-1} + H_i\ddot{Q}_i \qquad (22)$$

$$F_i = I_i\dot{V}_i + \hat{P}_i F_{i+1} \qquad (23)$$

which represent the simplified N-E algorithm (excluding the nonlinear terms) for the serial chain.

IV. Schur Complement Factorization of $\mathcal{M}^{-1}$

A. Interbody Force Decomposition Strategy

In this section we briefly review a recently developed factorization of $\mathcal{M}^{-1}$ [9,10] to establish the basis for developing a similar factorization for Λ^{-1} and Λ. This new factorization is based on a rather unconventional decomposition of interbody force of the form:

$$F_i = H_i F_{Ti} + W_i F_{Si} \qquad (24)$$

where F_{Si} is the constraint force. The projection matrices H_i and W_i are taken to satisfy the following orthogonality conditions:

$$H_i^t H_i = U,\ W_i^t W_i = U,\ W_i^t H_i = O \qquad (25)$$

$$H_i H_i^t + W_i W_i^t = U \qquad (26)$$

Note that the projection matrices are taken to be block diagonal in the rotational and translational coordinates. This implies that there is no coupling between the degrees of freedom, thereby precluding dimensional inconsistency (see [9] for a more detailed discussion.) For a joint i with multiple DOFs, say $l_i < 6$ DOFs, $H_i\epsilon\Re^{6\times l_i}$ and $W_i\epsilon\Re^{6\times(6-l_i)}$.

The decomposition in Eq. (24) naturally leads to the explicit computation of the constraint forces. In fact, researchers have often argued that since the constraint forces are nonworking forces their explicit evaluation, which leads to the computational inefficiency, should be avoided. Interestingly, however, the decomposition in Eq. (24) leads to new factorizations of $\mathcal{M}^{-1}$, Λ^{-1}, and Λ^{-1} and subsequent optimal serial and parallel algorithms.

B. Factorization of $\mathcal{M}^{-1}$

In [9,10], it has been shown that the force decomposition in Eq. (24) leads to a new Schur Complement factorization of $\mathcal{M}^{-1}$. Here, we briefly review this factorization since it is needed for derivation of the factorization of Λ^{-1} and Λ. To begin, let us define following global matrix and vector for $i = N$ to 1:

$$\mathcal{W} \triangleq \text{diag}\{W_i\}\epsilon\Re^{6N\times 5N};\ \mathcal{F}_S \triangleq \text{col}\{F_{Si}\}\epsilon\Re^{5N}$$

Equations (22)-(26) can be now written in global form as

$$\mathcal{P}^t\dot{\mathcal{V}} = \mathcal{H}\ddot{Q} \qquad (27)$$

$$\mathcal{P}\mathcal{F} = \mathcal{I}\dot{\mathcal{V}} \qquad (28)$$

$$\mathcal{F} = \mathcal{H}\mathcal{F}_T + \mathcal{W}\mathcal{F}_S \qquad (29)$$

$$\mathcal{H}^t\mathcal{H} = U,\ \mathcal{W}^t\mathcal{W} = U, \text{ and } \mathcal{W}^t\mathcal{H} = O \qquad (30)$$

$$\mathcal{H}\mathcal{H}^t + \mathcal{W}\mathcal{W}^t = \mathcal{U} \qquad (31)$$

From Eqs. (27), (28), and (30) it follows that

$$\dot{\mathcal{V}} = \mathcal{I}^{-1}\mathcal{P}\mathcal{F} \qquad (32)$$

$$\mathcal{W}^t\mathcal{P}^t\dot{\mathcal{V}} = \mathcal{W}^t\mathcal{H}\ddot{Q} = 0 \qquad (33)$$

and from Eqs. (32)-(33), we get

$$\mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{F} = 0 \qquad (34)$$

Substituting Eq. (29) into Eq. (34) yields

$$\begin{aligned} &\mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}(\mathcal{H}\mathcal{F}_T + \mathcal{W}\mathcal{F}_S) = 0 \Rightarrow \\ &\mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{W}\mathcal{F}_S = -\mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{H}\mathcal{F}_T \end{aligned} \qquad (35)$$

From Eqs. (35) and (29) it follows that

$$\mathcal{F} = (\mathcal{H} - \mathcal{W}(\mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{W})^{-1}\mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{H})\mathcal{F}_T \quad (36)$$

Multiplying both sides of Eq. (27) by $\mathcal{H}^t$ and from Eq. (30), Q is then computed as

$$\mathcal{H}^t\mathcal{H}\ddot{Q} = \mathcal{H}^t\mathcal{P}^t\dot{\mathcal{V}} \Rightarrow \mathbf{Q} = \mathcal{H}^t\mathcal{P}^t\dot{\mathcal{V}} \quad (37)$$

Finally, by computing $\dot{\mathcal{V}}$ from (32) and (36) and substituting it into (37), we get

$$\ddot{Q} = (\mathcal{H}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{H} - \mathcal{H}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{W} (\mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{W})^{-1}\mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{H})\mathcal{F}_T \quad (38)$$

In comparison with Eq. (21), an operator factorization of $\mathcal{M}^{-1}$, in terms of its decomposition into a set of simpler operators, is given by

$$\mathcal{M}^{-1} = \mathcal{C} - \mathcal{B}^t\mathcal{A}^{-1}\mathcal{B} \quad (39)$$

$$\mathcal{A} = \mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{W}\in\Re^{5N\times 5N}$$
$$\mathcal{B} = \mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{H}\in\Re^{5N\times N}$$
$$\mathcal{C} = \mathcal{H}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{H}\in\Re^{N\times N}$$

Note that, d and $\mathcal{B}$ are block tridiagonal matrices and C is a tridiagonal matrix. Also, both $\mathcal{A}$ and $\mathcal{C}$ are Symmetric Positive Definite (SPD) [10].

The operator form of $\mathcal{M}^{-1}$ given by Eq. (39) represents an interesting mathematical construct. If a matrix $\mathcal{L}_1$ is defined as

$$\mathcal{L}_1 = \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{B}^t & \mathcal{C} \end{bmatrix} \in\Re^{6N\times 6N}$$

then $\mathcal{M}^{-1}$ is the Schur Complement of $\mathcal{A}$ in $\mathcal{L}_1$. The structure of matrix $\mathcal{L}_1$ not only provides a deeper physical insight into the computation but it also motivates a different and a much simpler approach for derivation of the factorization of $\mathcal{M}^{-1}$ [10,11].

V. Schur Complement Factorization of Λ^{-1} and A

A. Schur Complement Factorization of Λ^{-1}

The factorization of $\mathcal{M}^{-1}$ directly results in a new factorization of Λ^{-1}. This factorization is derived by substituting the factorization of $\mathcal{J}$, given by (20), and $\mathcal{M}^{-1}$, given by (39), into (9):

$$\Lambda^{-1} = \beta(\mathcal{P}^t)^{-1}\mathcal{H}(\mathcal{H}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{H} - \mathcal{H}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{W} (\mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{W})^{-1}\mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{H})\mathcal{H}^t\mathcal{P}^{-1}\beta^t$$

which can be written as

$$\Lambda^{-1} = \beta(\mathcal{P}^t)^{-1}(\mathcal{H}\mathcal{H}^t)\mathcal{P}^t(\mathcal{I}^{-1} - \mathcal{I}^{-1}\mathcal{P}\mathcal{W} (\mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{W})^{-1} \mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1})\mathcal{P}(\mathcal{H}\mathcal{H}^t)\mathcal{P}^{-1}\beta^t \quad (40)$$

The key to simplification of this expression is the fact that, from Eq. (31), we have

$$\mathcal{H}\mathcal{H}^t = U - \mathcal{W}\mathcal{W}^t \quad (41)$$

By replacing Eq. (41) into Eq. (40) and after some involved algebraic manipulations, a simple operator expression of Λ^{-1} is derived as

$$\Lambda^{-1} = \beta\mathcal{I}^{-1}\beta^t - \beta\mathcal{I}^{-1}\mathcal{P}\mathcal{W}(\mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{W})^{-1} \mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\beta^t \quad (42)$$

This expression can be further simplified since

$$\mathcal{E}^t = \beta\mathcal{I}^{-1}\mathcal{P}\mathcal{W} = [\hat{P}_N^t I_N^{-1} W_N, 0, \dots 0]\in\Re^{6\times 5N} \quad (43)$$

$$\mathcal{D} = \beta\mathcal{I}^{-1}\beta^t = \hat{P}_N^t I_N^{-1}\hat{P}_N \quad (44)$$

The parallel axis theorem in Eq. (15) can be also used for propagation of the inverse of spatial inertia. By using Eq. (14), Eq. (44) can be written as

$$\mathcal{D} = ((\hat{P}_N)^{-1}(I_N)(\hat{P}_N^t)^{-1})^{-1} = (\hat{P}_{N+1,N} I_N \hat{P}_{N+1,N}^t)^{-1} = I_{N,N+1}^{-1} \quad (45)$$

that is, $\mathcal{D}$ is just the inverse of spatial inertia of link N about point O_{N+1}. The factorization of Λ^{-1} can be written in form of Schur Complement as

$$\Lambda^{-1} = \mathcal{D} - \mathcal{E}^t\mathcal{A}^{-1}\mathcal{E} \quad (46)$$

Note that the matrix $\mathcal{A}$ is the same as in Eq. (39). If a matrix $\mathcal{L}_2$ is defined as

$$\mathcal{L}_2 = \begin{bmatrix} \mathcal{A} & \mathcal{E} \\ \mathcal{E}^t & \mathcal{D} \end{bmatrix} \in\Re^{(5N+6)\times(5N+6)}$$

then Λ^{-1} is the Schur Complement of A in $\mathcal{L}_2$.

B. Schur Complement Factorization of Λ

Once Λ^{-1} is computed and assuming that its inverse exists (i.e., Λ^{-1} is nonsingular), Λ can be then obtained by a 6 x 6 matrix inversion. However, this corresponds to a numerical evaluation of Λ. Interestingly, it is possible to derive a factorization of Λ which allows its direct computation without any need for computing Λ^{-1}. This also provides a deeper physical insight into the structure as well as a simple physical interpretation of matrix Λ.

The factorization of Λ is derived by using the matrix identity

$$(C - XDY)^{-1} = C^{-1} - C^{-1}X(YC^{-1}X - D^{-1})^{-1}YC^{-1}$$

for inverting Λ^{-1}, given by Eq. (46), as

$$\Lambda = \mathcal{D}^{-1} - \mathcal{D}^{-1}\mathcal{E}^t\mathcal{S}^{-1}\mathcal{E}\mathcal{D}^{-1} \quad (47)$$

where $\mathcal{S} = \mathcal{E}\mathcal{D}^{-1}\mathcal{E}^t - \mathcal{A}$. This inversion, in addition to the nonsingularity of A^{-1}, also requires that the matrix $\mathcal{S}$ be nonsingular (note that, $\mathcal{D}$ is positive definite and hence $\mathcal{D}^{-1}$ exists.) It should be mentioned that there are other possible forms of A which only require the nonsingularity of A^{-1} [11]. The above expression of A can be further simplified by noting that

$$\mathcal{G} = \mathcal{D}^{-1} = (I_{N,N+1}^{-1})^{-1} = I_{N,N+1} \qquad (48)$$

Also, from Eqs. (43) and (45), we get

$$D\text{-}it' = [\hat{P}_N^{-1} I_N (\hat{P}_N^t)^{-1} \hat{P}_N^t I_N^{-1} W_N, 0, \ldots, 0]$$

$$\mathcal{R}^t = \mathcal{D}^{-1}\mathcal{E}^t = [\hat{P}_{N,N+1} W_N,\ 0,\ \ldots,\ 0] \epsilon \Re^{6\times 5N} \qquad (49)$$

Note that, from Eq. (49), we have

$$\mathcal{S} = \mathcal{E}\mathcal{R}^t - \mathcal{A} \qquad (50)$$

which implies that S is a rank one (in block sense) modification of matrix $\mathcal{A}$ i.e, $\mathcal{S}$ differs from $\mathcal{A}$ only in the leading element. The above factorization of A can be written in form of Schur Complement as

$$A = \mathcal{G} - \mathcal{R}^T \mathcal{S}^{-1} \mathcal{R} \qquad (51)$$

If a matrix $\mathcal{L}_3$ is defined as

$$\mathcal{L}_3 = \begin{bmatrix} \mathcal{S} & \mathcal{R} \\ \mathcal{R}^t & \mathcal{G} \end{bmatrix} \epsilon \Re^{(5N+6)\times(5N+6)}$$

then A is the Schur complement of S in $\mathcal{L}_3$.

VI. Complexity of Serial and Parallel Computation of A

A. $O(N)$ Serial Computation

For the sake of space, we only discuss the computation of A by explicit computation and inversion of A^{-1}. Note that, similar results can be also obtained by direct. computation of A from Eq. (51) and using Eq. (50). However, as will be discussed below explicit computation of A^{-1} rather than A provides a greater efficiency in both serial and parallel computation.

The most computation-intensive kernel in both operator application of M^{-1} and computation of A^{-1} is the computation and inversion of matrix A. The matrix A and its elements are given as

$$A = \text{Tridiag}[B_i, A_i, B_{i-1}^t]$$

$$A_i = W_i^t (I_i^{-1} + \hat{P}_{i-1}^t I_{i-1}^{-1} \hat{P}_{i-1}) W_i \quad i = N \text{ to } 1 \qquad (52)$$

$$B_i = -W_i^t I_i^{-1} \hat{P}_i W_{i+1} \quad i = N - 1 \text{ to } 1 \qquad (53)$$

From Eqs. (52)-(53) the elements of matrix A can be computed in O(N) steps. Efficient computation of matrix A by using optimal frame for projection of Eqs.(52)-(53) is discussed in [9].

The explicit computation of A^{-1} from Eq. (46) can be performed in O(N) steps as follows. The computation of A'& corresponds to solution of the system

$$\mathcal{A}\Omega = \mathcal{E} \qquad (54)$$

for Ω. This represents the solution of a SPD block tridiagonal system for six right-hand side vectors which, by using the block LDL^t algorithm, can be obtained in O(N) steps. Given the sparse structure of $\mathcal{E}^t$ (see Eq. (43)), the computation of $\mathcal{E}^t\Omega$ can be red uced to

$$\Theta = \mathcal{E}^t\Omega = \mathcal{E}_N^t \Omega_N \qquad (55)$$

where $\mathcal{E}_N^t \epsilon \Re^{6\times 5}$ and $\Omega_N \epsilon \Re^{5\times 6}$ are the first elements oft' and Ω (note the ordering of these matrices, e.g. $\mathcal{E}^t$ in Eq. (43).) The matrix-matrix multiplication in Eq. (55) can be performed with a cost of O(l). The matrix A^{-1} can be then obtained by adding two 6 x 6 matrices with a cost of O(l). Finally, A is obtained by a 6 x 6 matrix inversion, resulting in an O(N) complexity for the overall computation.

The most computation-intensive part of block LDL^t algorithm is the factorization of the block tridiagonal matrix since it requires the inversion of 5 x 5 matrices. However, for both the operator application of M^{-1} and computation of Λ^{-1} this factorization needs to be performed only once. This clearly demonstrates the synergism between the operator application of M^{-1} and computation of A^{-1} since once this factorization is obtained for operator application of M^{-1}, the computation of Eq. (54) can be performed with a much greater efficiency.

B. $O(Log\ N)$ Parallel Computation

The computation of elements of matrix A from Eqs. (52)-(53) is fully decoupled for i = N to 1. Thus, by using O(N) processors, this computation as well as required projections can be performed in 0(1) while involving only nearest neighbor communication among processors [9].

The block LDL^t algorithm, while highly efficient for serial solution of block tridigonal systems, seems to be strictly sequential and, in fact, there is no report on its parallelization. However, the Block Cyclic Reduction (BCR) algorithm [15], while less competitive for serial computation can be efficiently parallelized. By using the BCR algorithm, the system in Eq. (54) can be solved in $O(Log\ N)$ steps with O(N) processors. The computation of Eq. (55) and the final matrix addition for computation of A^{-1} as well as its inversion can be each performed in O(1) with one processor, i.e., in a serial fashion. This results in a complexity of $O(Log\ N)$ + O(1) for a parallel computation of A with O(N) processors which indicates a both time- and processor-optimal parallel algorithm.

It should be emphasized that efficient parallel solution of block tridiagonal systems is the key to an efficient parallel computation of Schur Complement factorization of M^{-1}, A^{-1} and Λ. Motivated by this fact, we have developed a more efficient variant of the BCR algorithm [21 ,22] which is particularly uitable for implementation on coarse grain MIMD prallel architect ure since it significant ly reduces the communication overhead by providing a high degree of overlapping between communication and computation [16]. However, an even further efficiency in parallel implementation of OSC scheme can be achieved by exploiting the synergism in operator application of M^{-1} and the computation of A-l. In the operator application of M^{-1} the multiplication of $\mathcal{A}^{-1}$ by a vector is needed. This corresponds to the solution of a system as:

$$\mathcal{A}X = Y \tag{56}$$

for some vectors X and Y. Now, (54) and (56) can be combined and solved as a linear system with seven right-hand side vectors. This combination not only increase the grain size but also reduces the amount of communication in the parallel Implemental.lon.

VII. Discussion and Conclusion

We presented a new factorization technique for computation of Λ^{-1} and A. This technique results in Schur Complement factorization of both Λ^{-1} and A and subsequently new O(N) algorithms for their computation. These $O(N)$ algorithms are highly efficient for parallel computation. To our knowledge, they re pesent the first algorithms that can be fully parallelized, resulting m both time- and processor-optimal parallel algorithms. Using these algorithms, the OSC scheme can be implemented with optimal efficiency in both serial and parallel environment.

However, in addition to their theoretical significance, these algorithms are also highly efficient for practical implementation on MIMD parallel architectures. We have implemented the parallel $O(Log\,N)$ algorithm for computation of forward dynamics of a serial chain by using the Schur Complement factorization of M^{-1} on a Hypercube architecture [16]. Our results clearly validate the efficiency of this factorization of M^{-1} for practical parallel implementation. However, as discussed in §VI.A, one can expects an even gester efficiency in the parallel implementation of the OSC scheme.

The manifest of Schur Complement in factorization of M^{-1}, A^{-1} and A provides a unified framework not only for their computations but also for their physical interpretations [1 1]. In fact, we strongly believe that the physical insight provided by these factorization can lead to a better understanding of the control schemes. The exploitation of this physical insight in the design of task space control schemes as well as the practical real-time, parallel implementation of the OSC scheme by using the algorithms of this paper will be discussed in forthcoming reports.

ACKNOWLEDGMENT

The research of first author was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration (NASA).

References

1. 0. Khatib," A Unified Approach for Motion and Force Control of Robot. Manipulators: The Operational Space Formulation," *IEEE J. Robotics & Automation*, Vol. 3(l), 1987.
2. O. Khatib," Inertia Properties in Robotic Manipulation: An Object Level Framework," *Int. J. Robotics Research*, Vol. 13(1), pp. 19-36, 1995.
3. J.Y.S. Luh, M.W. Walker, and R.P.C. Paul,"On-Line Computational Scheme for Mechanical Manipulator," *ASME J. Dynamic Syst., Mess., Control*, Vol. 102, pp. 69-76, June 1980.
4. K.W. Lilly and D.E. Orin," Efficient $O(N)$ Computation of the Operational Space Inertia Matrix," *Proc. IEEE Int. Conf. Robotics & Automation*, pp. 1014-1019, Cincinnati, OH, May 1990.
5. I{. Kreutz-Delgado, A. Jain, and G. Rodriguez, "Recursive Formulation of Operational Space Control," *Int. J. Robotics Res.*, Vol. 11(4), pp. 320-328, Aug. 1992.
6. G. Rodriguez and K. Kreutz-Delgado, "Spatial Operator Factorization and Inversion of the Manipulator Mass Matrix," *IEEE Trans. Robotics & Automation, Vol. RA-8(1), pp.* 65-76, *Feb. 1992.*
7. *R. Featherstone, "The* Calculation *of Robot Dynamics* Using *Articulated-Body Inertia, " Int. J. Robotics Res., Vol. 2(l), pp. 13-90, 1989.*
8. *A. Fijany and A. K. Bejczy, "Techniques for Parallel* Computation *of Mechanical* Manipulator *Dynamics. Part II: Forward Dynamics, "* in *Advances in Control and Dynamic ystems, Vol.* 40, *C. T., Leondes (Ed.), pp. 357-410, Academic Press, March* 1991.
9. *A. Fijany, I. Sharf, and G. D'Eleuterio, "Parallel O(Log N) Algorithms for Computation of* Manipulator *Forward* Dynamics, *" IEEE Trans. Robotics & Automation, Vol. 11(.9), pp. 389-400,* June *1995.*
10. *A. Fijany, "Parallel O(log N) Algorithms jor* Rigid *Multibody Dynamics, " JPL* Engineering Memorandum, *EM 343-92-1258, Aug. 1992.*
11. *A. Fijany,"Schur Complement* Factorization: *A New Framework jor Manipulator* Dynamic Analysis, Control, *and* Simulation, *" In preparation.*
12. *A. Fijany and A .K. Bejczy, "ASPA RC: An Algorithmically* Specialized *Parallel Architecture for Robotics Computations, "* in *Parallel Computation Systems for obotics: Algorithms and Architectures, A. Fijany and A .K. Bejczy (Eds.), World* Scientific *Pub., 1992.*
19. *J. Angeles, "On* Numerical Solution *of* Inverse Kinematic *Problem, " Int. J. Robotic Res., Vol. 4(2), pp. 21-37, 1985.*
14. *C.S. G. Lee and P.R. Chang,"Efficient Parallel Algorithms for Robot Inverse* Dynamics *Computation, " IEEE Trans. Syst., Man, and Cybern., Vol. 16(4), pp. 5S2-542, July/August 1986.*
15. *R. W. Hackney and C.R. Jesshope,* Parallel Computers, Adam *Hilger Ltd, 1981.*
16. *A. Fijany,* G. Kwan, *and N. Bagherzadeh, "A Fast Algorithm for* Parallel *Computation of Multibody* Dynamics on *MIMD Parallel Architectures, " Proc. Computing in Aerospace 9 Conf. pp. 1021-1032, San* Diego, *CA, Oct. 1993.*

Nomenclature

N	Number of Degrees-Of-Ileedom (DOF) of the system
$p_{i,j}$	Position vector from Oj to O_i, $p_{i+1,i} = p_i$
m_i	Mass of **link i**
J_i	**Second moment of mass of link i about its center of mass**
h_i, k_i	**First** and Second Moment of mass of link i about point O_i
$Q_i, \dot{Q}_i, \ddot{Q}_i$	Position, velocity, and acceleration of joint i
Γ_i	Applied (control) force on joint i
$\omega_i, \dot{\omega}_i \epsilon \Re^3$	Angular velocity and acceleration of link i
$v_i, \dot{v}_i \epsilon \Re^3$	Linear velocity and acceleration of link i, point O_i
$f_i, n_i \epsilon \Re^3$	Force and moment of interaction between link i-1 and link i
H_i	Spatial axis (map matrix) of joint i, $H_i \epsilon \Re^{6\times k}$ for a joint with k DOFS
$I_{i,j} \epsilon \Re^{6\times 6}$	Spatial Inertia of body i about point Oj, $I_{i,i} = I_i$

$$I_i = \begin{bmatrix} k_i & h_i \\ \tilde{h}^t & m_i U \end{bmatrix}$$

(t denotes transpose)

$V_i = \begin{bmatrix} \omega_i \\ v_i \end{bmatrix} \epsilon \Re^6$	Spatial velocity of link i, point O_i
$\dot{V}_i = \begin{bmatrix} \dot{\omega}_i \\ \dot{v}_i \end{bmatrix} \epsilon \Re^6$	Spatial acceleration of link i, point O_i
$F_i = \begin{bmatrix} n_i \\ f_i \end{bmatrix} \epsilon \Re^6$	Spatial force of interaction between link **i-1 and link i**
$F_{N+1} \epsilon \Re^6$	**External** spatial force acting on the End-Effecter (EE)
$V_{N+1}, \dot{V}_{N+1} \epsilon \Re^6$	EE Spatial velocity and acceleration, point O_{N+1}

Global Quantities, i = N to 1

$\mathcal{M} \epsilon \Re^{N\times N}$	Symmetric Positive Definite (SPD) mass matrix
$\mathcal{J} \epsilon \Re^{6\times N}$	Jacobian Matrix
$\mathcal{H} = \mathrm{Diag}\{H_i\}$	Global matrix of spatial axes, $\mathcal{H} \epsilon \Re^{6N\times N}$ for a system with 1 DOF joints.
$\mathcal{I} = \mathrm{Diag}\{I_i\} \epsilon \Re^{6N\times 6N}$	Global matrix of spatial inertia
$Q = \mathrm{Col}\{Q_i\} \epsilon \Re^N$	Global Vector of joint positions
$\dot{Q} = \mathrm{Col}\{\dot{Q}_i\} \epsilon \Re^N$	Global vector of joint velocities
$Q = \mathrm{Col}\{\ddot{Q}_i\} \epsilon \Re^N$	Global vector of joint accelerations
$r = \mathrm{Col}\{\Gamma_i\} \epsilon \Re^N$	**Global** vector of applied joint forces
$V = \mathrm{Col}\{V_i\} \epsilon \Re^{6N}$	Global vector of spatial velocities
$V = \mathrm{Col}\{\dot{V}_i\} \epsilon \Re^{6N}$	Global vector of spatial accelerations
$\mathcal{F} = \mathrm{Col}\{F_i\} \epsilon \Re^{6N}$	Global vector of spatial interaction forces

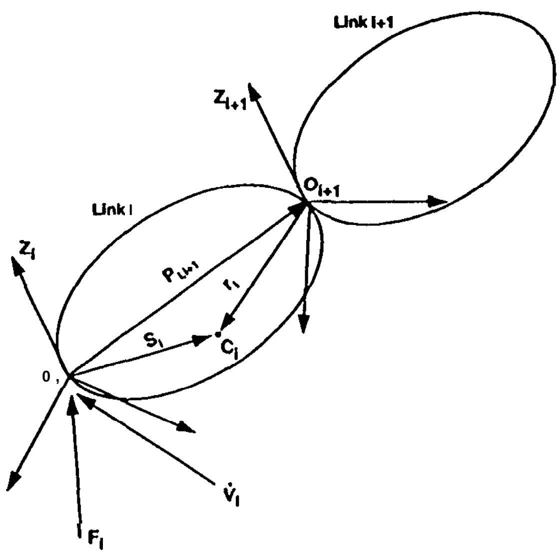

Figure 1. Links, Frames, and Position Vectors